Incident Report: Suspicious Web Activity and Potential Attack

Incident Report: Suspicious Web Activity and Potential Attack

Introduction

This report outlines a series of suspicious activities observed within the server logs, with the aim to analyze potential security incidents, determine their nature, and propose mitigation strategies. The incidents in question involve a high number of POST requests from one IP address, accompanied by a series of requests with suspicious referrers. The investigation suggests potential bot activity, coupled with proxy or redirect attempts, which may indicate automated attacks or attempts to exploit vulnerabilities in the web application.

Detailed Attack Analysis

IP 1 (5000 Requests)

- **IP Address**: `54.145.34.34`
- **Activity**: Generated 5000 POST requests to the server within a 2-minute period.
- **User-Agent**: `python-requests/2.31.0`
- **Impact**: The sheer volume of requests from a single IP address in a short period suggests automated activity, potentially a bot attempting to flood the server or carry out a brute-force attack, targeting a specific endpoint or testing for vulnerabilities such as login, rate-limiting thresholds, or server stability.

IP 2 (11 Requests with Referrer IP)

- **IP Address**: `51.158.115.139`
- **Activity**: Generated 11 POST requests to the server.
  - Of these, 3 requests had another IP address in the **referrer** field: `http://75.101.231.9:4001/`.
  - `51.158.115.139 - - [14/Jun/2024:17:28:55 +0000] "POST / HTTP/1.1" 200 1941 "http://75.101.231.9:4001/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.0.0 Safari/537.36" "-"`
- **User-Agent**: `Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.0.0 Safari/537.36`

- **Impact**: The inclusion of an external referrer IP suggests the possibility of an attempt to **obfuscate the true origin** of the requests, indicating potential **proxying**, **man-in-the-middle (MitM)** attacks, or the use of an **open redirect** service. This could be an attempt to exploit vulnerabilities, possibly to launch attacks like **session hijacking** or **phishing**.

Referrer Analysis

- The referrer IP `http://75.101.231.9:4001/` raises a red flag, suggesting that it could be a **proxy server**, a **VPN**, or an external malicious server used to disguise the origin of the requests.
- The presence of this IP in the referrer field, along with **unsuspecting traffic** following, points to the potential exploitation of **open redirect vulnerabilities**.

Proposed Mitigation Strategy

To mitigate future attacks and reduce the likelihood of such incidents, the following strategies are recommended:

1. **Implement Rate Limiting**:
    - Apply **rate limiting** and **CAPTCHA** mechanisms to endpoints receiving a high volume of requests (e.g., login forms or POST endpoints) to prevent abuse and **automated bot activity**.
    - Block IPs generating requests at an unusually high rate.
2. **Inspect and Sanitize Referrer Field**:
    - Sanitize and **validate incoming referrer data** to block malicious referrers or external redirects that might bypass the server's security.
    - Prevent or limit redirection to external URLs to mitigate **open redirect vulnerabilities**.
3. **Block Known Malicious IPs**:
    - Investigate the **referrer IP** (`75.101.231.9`) to determine if it is part of a known malicious network or proxy service. Consider blocking suspicious IP addresses or **throttling requests** from unfamiliar or untrusted sources.
4. **Apply Web Application Firewalls (WAF)**:
    - Deploy a **Web Application Firewall (WAF)** to detect and block malicious traffic based on patterns such as high-frequency requests, suspicious referrers, or unusual headers.
5. **Monitor and Analyze Logs**:
    - Set up **real-time monitoring** for server logs to identify and respond to high-volume request patterns or unexpected traffic originating from unusual referrers.
    - Regularly analyze logs for evidence of attack attempts, such as **DoS/DDoS** or **brute-force** activities, as well as **suspicious redirects**.

Justification for the Proposed Solution

The suggested mitigation strategies align with industry standards for defending against **DoS**, **bot attacks**, and **man-in-the-middle** scenarios. These measures are proven to:

- **Limit traffic** from malicious actors by blocking or slowing down requests from suspicious sources.
- **Prevent exploitation of web application vulnerabilities**, including open redirects and the abuse of external referrers.
- **Increase the server's resilience** to automated attacks by using CAPTCHA and rate-limiting features.

Steps for Implementation

1. **Configure Rate Limiting**:
    - Set up rate limiting on the server to allow only a certain number of requests from each IP per minute/hour.
    - Integrate CAPTCHA or other bot detection mechanisms on sensitive forms.
2. **Review and Sanitize Referrer Headers**:
    - Implement checks for suspicious or unauthorized referrers and reject or sanitize the header accordingly.
    - If possible, configure the application to avoid accepting external URLs in referrer fields.
3. **Block or Throttle Suspicious IPs**:
    - Use a **IP reputation service** to check and block known malicious IP addresses.
    - Implement dynamic blocking of IPs based on behavior, such as repeated requests from the same IP address.
4. **Deploy a WAF**:
    - Implement a **Web Application Firewall** to actively inspect incoming traffic and block malicious requests before they reach the server.
5. **Set Up Continuous Monitoring**:
    - Use monitoring tools (such as **Splunk**, **ELK Stack**, or **Datadog**) to track real-time server logs for any signs of attack patterns or abnormal traffic.

Post-Implementation Monitoring

Post-implementation, it is critical to:

- **Monitor traffic** regularly to ensure that the mitigation measures are working and to catch any new suspicious activities early.
- **Perform regular security audits** on the web application, focusing on traffic patterns and referrer-related exploits.
- **Review the effectiveness** of CAPTCHA, rate limiting, and other protections periodically and update them as necessary.

Conclusion

This report analyzed suspicious activity observed on the server, which indicates potential **bot-driven attacks**, **proxy usage**, and exploitation of **open redirect vulnerabilities**. By implementing the proposed mitigation strategies, such as rate limiting, referrer sanitization, and web application firewall deployment, the organization can significantly reduce the risk of future attacks. Ongoing monitoring and log analysis will ensure early detection of new threats and the continued protection of critical systems.
